

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-233510
 (43)Date of publication of application : 10.09.1993

(51)Int.CI. G06F 13/00
 G06F 3/153
 G06F 15/00
 G09G 5/22
 G09G 5/24

(21)Application number : 04-037734
 (22)Date of filing : 25.02.1992

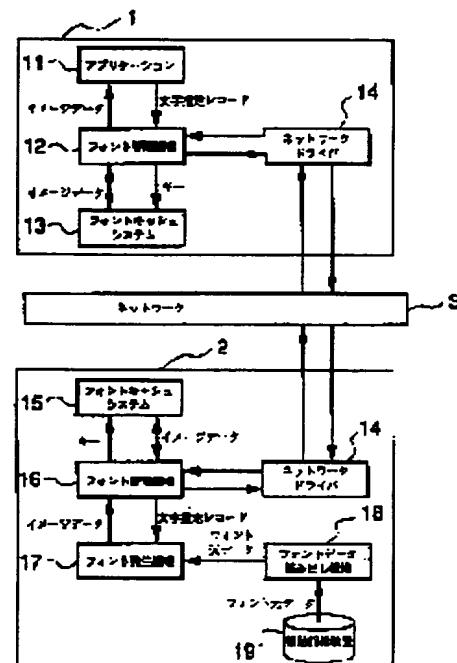
(71)Applicant : SEIKO EPSON CORP
 (72)Inventor : NAGASAKA FUMIO

(54) CHARACTER DATA CONTROL METHOD

(57)Abstract:

PURPOSE: To generate the character data at a high speed by providing the font cache memories to both a character output requesting device and a character image data supply device which are included in a network.

CONSTITUTION: An exclusive device (font server) 2 is connected to a network 3 to generate the character image data from the original data on an outline font. These character image data are supplied to a device which outputs the characters in the network 3 from the server 2. Then a character requesting computer 1 and the server 2 are provided with the memory areas, i.e., the font caches which store the character image data. At the same time, the control is carried out so as to keep the discordance of contents between the cache memory of the server 2 and that of the computer 1. Then the character data which are once converted into the image data are repetitively used as much as possible. As a result, the processing frequency can be reduced for generation of the character data from the original data on the outline font.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

BEST AVAILABLE COPY

[of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-233510

(43)公開日 平成5年(1993)9月10日

(51)IntCl.⁵
G 0 6 F 13/00 3 5 7 Z 7368-5B
3/153 3 1 0 B 9188-5B
15/00 3 1 0 R 7459-5L
G 0 9 G 5/22 9061-5G
5/24 9061-5G

F I

技術表示箇所

審査請求 未請求 請求項の数1(全24頁)

(21)出願番号 特願平4-37734

(22)出願日 平成4年(1992)2月25日

(71)出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72)発明者 長坂 文夫

長野県諏訪市大和3丁目3番5号セイコーエプソン株式会社内

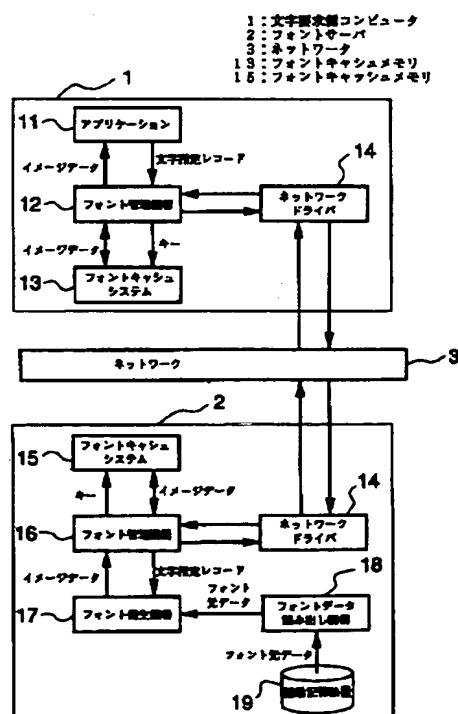
(74)代理人 弁理士 鈴木 喜三郎 (外1名)

(54)【発明の名称】 文字データ管理方法

(57)【要約】

【目的】 ネットワーク上の、文字出力要求側装置と文字のイメージデータ供給装置の双方にフォントキッシュメモリを配置することで、文字データ生成を高速に処理する。

【構成】 アウトラインフォントの元データから、文字のイメージデータを生成する処理を行なう専用装置（フォントサーバ）をネットワークに接続し、ネットワーク上の文字出力を行なう装置は、フォントサーバから文字のイメージデータの供給を受ける。ここで、要求側コンピュータと、フォントサーバの両者にそれぞれフォントキッシュと呼ぶ、文字のイメージデータを保持するメモリ領域を設け、かつフォントサーバのキッシュメモリの内容と、文字要求側コンピュータのキッシュメモリの内容を不一致に保つ制御を行ない、一度イメージデータに変換された文字データをできるだけ繰り返し利用し、アウトライン元データからの生成処理の実行を減らすことを特徴としている。



【特許請求の範囲】

【請求項 1】 文字出力動作を行う複数台の装置と、文字データ資源を持つ1台以上の装置が、ネットワークにより接続されたシステムで使用され、
 文字出力動作を行う装置については、出力すべき文字の文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の形状を表すデータが、指定された特定のメモリ領域（以下ローカルフォントキャッシュと書く）に存在することを検索する手段と、この検索により該当する文字のデータが有った場合、文字出力要求を発生した処理に文字データを渡す手段と、この検索により該当する文字が無かった場合、ネットワーク上の予め定めた文字データ資源を持つ装置（以下フォントサーバと書く）に対し、文字の文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の形状を表す画素のデータを転送するよう促す要求メッセージを転送し、応答として得られる文字のデータを受け取る手段を持ち、応答として得られた文字のデータは、ローカルフォントキャッシュに登録を試み、ローカルフォントキャッシュに空き領域がない場合は、不要となった文字データを取り出し、空き領域を作り登録し、一方、不要として取り出した文字データは、フォントサーバに対し、同装置上の予め指定した特定のメモリ領域（以下グローバルフォントキャッシュと書く）に登録するよう促す要求メッセージを付加し、転送する手段を持つ構成であり、
 ネットワーク上のフォントサーバについては、ネットワーク上の他の装置より、文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の、文字取得要求を受信する手段と、この手段により受信された文字について、形状を表すデータが、指定された特定のメモリ領域に存在することを検索する手段と、この検索により該当する文字のデータが有った場合、このデータを読み取り、文字出力要求を発生した情報機器に渡す手段と、前記検索により該当する文字が無かった場合、指定された文字の形状を表す画素データを、アウトラインフォントデータより演算処理により発生し、前記の文字取得要求を発生した装置に、この画素データを、送信する手段を持ち、また、ネットワーク上の他の装置から文字データの登録を要求するメッセージを受信した場合は、グローバルフォントキャッシュへ登録を試み、空き領域が得られない場合、不要となった文字データを削除し、空き領域を作り前記登録を行なう手段を持つ構成からなることを特徴とする文字データ管理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数台のパソコンコンピュータ、ワークステーションをネットワークにより結合したシステムでの、アウトラインフォントの高速発生方式に関する。また、フォント形成に使用される画素

を、一時的に蓄えるメモリ構造（いわゆるフォントキャッシュ）の管理方式に関する。

【0002】

【従来の技術】 従来の装置として、特開平2-58094で記載された装置がある。この装置構成においては、フォントサーバと呼ぶ文字データ管理装置をネットワークに接続し、この同一のネットワーク上に、ワークステーション、印刷装置などを接続する。ここで、ワークステーションあるいは印刷装置は、文字出力を伴うアプリケーションプログラム（印刷装置の場合は印刷制御プログラム）が動作するものであり、出力するための文字データをフォントサーバに要求するクライアントである。また、フォント・キャッシュと呼ぶ機構が使用されている。フォント・キャッシュは、文字の形状データが、アウトライン・フォント形式で記録されている場合等に、処理の高速化を図るうえで効果的な機構である。アウトライン・フォントは、文字形状の元データが、文字形状を構成する輪郭を表す適当な曲線（例えばベジエ曲線、放物線等）の係数項や、曲線種類の指定などの情報を含むデータ列として記述されている。従って表示、印刷等、何らかの出力デバイスに文字として出力する場合は、元データから大きさの指定に従い画素データの集合へ変換しなければならない。しかしこの変換処理は、一般に処理時間を多く消費する。文字形状の複雑な漢字の場合、これは一層顕著である。そこで、一度画素に変換済みの文字データに関しては、変換済みの画素データをメモリに一時蓄えておき、同一文字に対し出力要求が発生した場合、元データからの画素への変換を行なうことなく、このメモリに既に記録されている画素データを使用する方法が採られる。ここで、画素に変換された文字形状のデータを蓄積するメモリが、フォントキャッシュと呼ばれる。特開平2-58094で記載された装置では、フォントサーバとネットワークによって接続されたワークステーション、印刷装置などの文字出力を行う装置側（すなわちクライアント側）に、フォント・キャッシュの機構が実現されている。このような従来方式の利点は主として次の3つである。

（1）文字データをネットワーク上のすべての装置が共有できるため、文字データ記憶のための補助記憶装置など資源が有効に利用できる。

【0003】（2）文字データが集中的に管理できるため、文字データ変更、追加、削除などの操作が簡単で保守性に優れる。

【0004】（3）文字データが特殊な処理を必要とする場合でも、フォントサーバにおいてのみ処理を行うだけで良い。

【0005】これに対し、従来方法の問題点は以下の点である。

【0006】（1）フォントサーバがクライアントマシンに対する唯一の文字データ供給源であり、文字出力

は、局所的なフォント・キャッシュが有効に働かない場合、フォントサーバの文字データ供給能力に律速される。

【0007】もしクライアントマシンが、表示フォントの文字種を切り替えたとすると、現在フォント・キャッシュにある文字データが再利用できない。従って、それ以降の文字出力については、しばらくの間、文字毎にフォントサーバへ要求が発生することになる。ここで、同時に複数のクライアントマシンが表示文字種の変更を行えば、フォントサーバへの要求が集中することになる。前述した様にアウトラインフォント等では、文字発生はプロセッサにとって大きな負荷である。複数の機器の要求に応じることは、ネットワークに付随する処理プログラムの負荷と併せ、処理速度の低下を引き起こす原因となる。

【0008】フォントキャッシュの利用率を高め、クライアントマシンでの局所的な応答を高めることは、キャッシュ管理方法の改善によって実現できる。例えば、フォントキャッシュ管理方法の効率向上を目的とし発明された従来技術に、特開昭64-88660"フォントキャッシュ制御方式"がある。しかし、ネットワークでの利用を前提とした、フォントサーバシステムの処理速度を向上するためには、ボトルネックとなる部分の解決が必要であり、クライアントマシンでの局所的な応答速度が向上しても、文字出力の最終的な速度の向上に結びつかないという問題点が有った。

【0009】

【発明が解決しようとする課題】以上に述べた様に、ネットワークでのフォントサーバの使用においては、フォントサーバの処理速度を可能な限り効果的に利用する技術を開発することが課題となる。これを解決する一つの方法は、フォントサーバにもフォントキャッシュを実装し、フォントサーバの処理速度を一層高める方法である。しかし、この方法を用いる場合、ワークステーション、印刷装置等のクライアントになるマシンのローカルなフォントキャッシュの内容と、フォントサーバのフォントキャッシュの内容との関係をどのように維持するかが問題となる。フォントサーバ上のフォントキャッシュはいわばネットワーク上でのグローバルなキャッシュであり、クライアントマシン側のローカルなキャッシュに有ると同一の文字データを保持する必要は無い。むしろ、ローカルなキャッシュとグローバルなキャッシュの内容に同一のデータ存在するのは、有限のキャッシュメモリの利用率の点から考え不都合である。そこで、フォントサーバ側のフォントキャッシュと、各クライアントマシンのフォントキャッシュの内容を、いかにして不一致に保つかという課題がある。

【0010】

【課題を解決するための手段】以上の課題を解決するため、本発明は、文字出力動作を行う複数台の装置と、文

字データ資源を持つ1台以上の装置が、ネットワークにより接続されたシステムで使用され、文字出力動作を行う装置については、出力すべき文字の文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の形状を表すデータが、指定された特定のメモリ領域（以下ローカルフォントキャッシュと書く）に存在することを検索する手段と、この検索により該当する文字のデータが有った場合、文字出力要求を発生した処理に文字データを渡す手段と、この検索により該当する文字が無かった場合、ネットワーク上の予め定めた文字データ資源を持つ装置（以下フォントサーバと書く）に対し、文字の文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の形状を表す画素のデータを転送するよう促す要求メッセージを転送し、応答として得られる文字のデータを受け取る手段を持ち、応答として得られた文字のデータは、ローカルフォントキャッシュに登録を試み、ローカルフォントキャッシュに空き領域がない場合は、不要となった文字データを取り出し、空き領域を作り登録し、一方、不要として取り出した文字データは、フォントサーバに対し、同装置上の予め指定した特定のメモリ領域（以下グローバルフォントキャッシュと書く）に登録するよう促す要求メッセージを付加し、転送する手段を持つ構成であり、ネットワーク上の他の装置より、文字種、文字修飾指定、文字サイズ指定、文字コード指定によって記述される文字の、文字取得要求を受信する手段と、この手段により受信された文字について、形状を表すデータが、指定された特定のメモリ領域に存在することを検索する手段と、この検索により該当する文字のデータが有った場合、このデータを読み取り、文字出力要求を発生した情報機器に渡す手段と、前記検索により該当する文字が無かつた場合、指定された文字の形状を表す画素データを、アウトラインフォントデータより演算処理により発生し、前記の文字取得要求を発生した装置に、この画素データを、送信する手段を持ち、また、ネットワーク上の他の装置から文字データの登録を要求するメッセージを受信した場合は、グローバルフォントキャッシュへ登録を試み、空き領域が得られない場合、不要となった文字データを削除し、空き領域を作り前記登録を行なう手段を持つ構成からなることを特徴とする。

【0011】

【実施例】図1はこの発明の一実施例のブロック構成図である。ネットワーク3は複数の装置間の双方向の高速な通信を確立する機構である。ネットワーク3に接続される複数の装置は、それぞれ固有のネットワーク・アドレスを持つ。装置間の通信は、送信側装置、受信側装置それぞれのネットワーク・アドレスを含むデータ列の転送により行われる。通常このデータ列はパケットと呼ばれる。各装置は、パケット内のデータ列中の、転送先

アドレスが自己的アドレスであった場合、パケットを取得する。また、パケット内のデータ列から、パケットを発信した装置のアドレスを取り出し、応答をそのアドレスの装置あてに返信する。これら機構の実現例としては、IEEE標準802などが広く使用されている。

【0012】図1の構成では、ネットワーク3に、装置1及び装置2が接続されている。言うまでもなく実際に更に多くの装置がネットワーク3に接続され使用される。ここで、装置1は、文字出力を要求するコンピュータであり、装置2はフォントサーバである。

【0013】(a) 文字出力要求側の動作の概要
まず、文字出力を要求するコンピュータ1について、その動作の概要を説明する。

【0014】コンピュータ1では、アプリケーション11が使用されている。アプリケーション11が文字の表示を行う場合、文字は、表示装置の解像度に合わせ生成された画素の集合として表現される。文字形状を表す画素の集合を以下、文字のイメージデータと書く。表示に使用する文字のイメージデータは、コンピュータ1が用意するアプリケーションインターフェースによって、アプリケーション11に供給される。フォント管理機構12はアプリケーションインターフェースの一部であり、アプリケーション11から文字指定レコードを受け取り、文字のイメージデータを11に返す動作を行う。ここで、文字指定レコードは以下の要素からなるデータ構造である。

【0015】

データ構造の名称： 文字指定レコード

要素1： 文字コード 2バイト

要素2： 書体指定 2バイト

要素3： 文字サイズ指定 2バイト

要素4： 文字修飾指定 6バイト

但し、書体は、ある文字について、複数の文字デザインが行われた場合、文字形状の差を特定する上で使用される呼称である。上記データ構造では、この呼称に数値を割り当てて使用する。また、文字修飾指定は、文字書体によらず、対象の文字の形状を、斜体文字、白抜き文字などの操作対象とする指定であり、書体と同様に、上記データ構造で使用する場合の内部表現は数値化されている。

【0016】フォント管理機構12は、文字指定レコードを受け取り、文字のイメージデータを11に返す動作において、フォントキャッシュシステム13へアクセスを行う。この動作の説明は後述する。

【0017】(b) 文字イメージデータ供給側の動作の概要

フォントサーバ2について、その動作の概要を図1の構成図を用いて説明する。

【0018】フォントサーバ2に対しネットワーク3に接続された他の装置は、文字のイメージデータの転送を

要求するパケットを渡す。この要求を受信したフォントサーバ2は、フォント管理機構16に要求を伝達する。フォント管理機構16は、要求のデータ列から、文字指定レコードを取り出し、後述の手段により指定に一致する文字のイメージデータがフォントキャッシュシステム15に存在するか調べる。存在する場合、文字指定レコードをキーに変換しキー指定によりイメージデータを取得し、要求を発生した装置に転送する。データが存在しない場合、フォント発生機構17にフォント発生を要求する。これを受けたフォント発生機構17は、フォントデータ読み出し機構18を使用し、補助記憶装置19から、フォント元データを読み出す。読み出されたフォント元データは、アウトラインフォントとしてフォントを記述したものである。フォント発生機構17は、このデータに基づき、文字毎にイメージデータを生成する。生成されたイメージデータは、フォント管理機構16に送られ、要求を発生した装置に転送される。

【0019】(c) フォント管理動作

以下に、ネットワーク上のフォントサーバと、要求発生を行う装置の関連からみた、フォント管理動作の説明を行う。

【0020】フォント管理機構が使用するフォントキャッシュメモリ管理方法として、いくつか公知の方法が存在する。基本的にキャッシュメモリは、有限長のメモリ割り当てであり、新たに文字のイメージデータを登録しようとし、メモリの空きスペースが確保できない場合は、現在の登録を更新する。すなわち、キャッシュ領域から、不要なデータ（ここではどれかの文字のイメージデータ）を削除し、新規のデータを登録する。管理方法として重要なのは、必要なデータを残し、不要なデータを削除することである。しかし、将来どのような文字データに対する要求が発生するかを予測する理想的な方法が無いため、何らかの基準に従い不要なデータを決める事になる。この決定が適切であれば、要求文字のデータが、キャッシュ領域に存在する確率（ヒット率）が向上し、処理速度の向上につながる。不要なデータを決める代表的な管理方法は以下の4つである。

【0021】方法1：現在のデータ群の中で、最も初期に登録されたデータが不要と判断される。いわゆるFirst In First Out(FIFO)の方法。

【0022】方法2：現在のデータ群の中で、最も長い期間使用されなかったデータが不要と判断される。いわゆるLeast Recently Used(LRU)の方法。

【0023】方法3：現在のデータ群の中で、最も長い期間使用されなかったデータはむしろ次に使用され、最近使用されたデータが不要と判断される方法。

【0024】方法4：現在のデータ群の中で、最も使用頻度の低いデータが不要と判断される。いわゆるLeast Frequently Used(LFU)の方法。

【0025】従来方法の内、例えば特開昭64-886

60" フォントキャッシュ制御方式" は方法1に対し方法3の改善を行った方法を用いている。また特開平2-202464"印刷装置" は方法1を用いている。

【0026】これら管理方式は、单一のフォント管理機構に対し、单一のフォントキャッシュメモリを使用する構成に対し発明された方法である。（より正確に言えば、仮想記憶方式でのメモリページ割り当てに関し発明された方法を、フォントキャッシュの管理に転用しただけである。）従って本実施例の様に、コンピュータ1のフォントキャッシュシステム13と、フォントサーバ2のフォントキャッシュシステム15が多重に存在する場合、最適なアルゴリズムとは言えない。既に述べた様に本発明においては、フォントサーバ側のフォントキャッシュと、各クライアントマシンのフォントキャッシュの内容を、不一致に保つ操作を行なう。

【0027】また、本実施例は、キャッシングの局所的な管理においてはLeast Recently Used(LRU)の方法を用いる。

【0028】(d) フォントキャッシュ管理方法
フォントキャッシュ管理方法について説明する。

【0029】(d1) フォントキャッシュシステム
図2はフォントキャッシュシステム13の構成図である。フォントキャッシュシステム15の構成も図2と同様である。但し既に述べた様に、フォントサーバ2においては管理方法が異なるため、別の番号により区別した。

【0030】フォントキャッシュシステム13の主たる構成要素はキャッシングメモリ205及びキャッシング割り当て処理207である。キャッシングメモリ205は有限長の連続なアドレス配置を持つメモリである。内部は128バイトの大きさの管理単位メモリ201の集合である。管理単位メモリ201は、大きさ126バイトのデータ部204と、大きさ2バイトの指標部203からなる。指標部203の内容は、次に続くべき管理単位メモリへの指標である。管理単位メモリ201は大きさ固定であるため、指標の値から、実アドレスの値は簡単な演算により求めることができる。すなわち指標の値をixとしたとき、次に続くべき管理単位メモリ201の先頭番地は、次式の値addrとして得られる。

【0031】

$addr = ix \times 128 + メモリ205\text{先頭番地}$

言うまでもなく、値128は管理単位メモリの大きさからくるもので、この演算は実アドレスへの変換の一例に過ぎない。管理単位メモリ210の指標部203は、いわゆる連結ポインタの変形の一つである。

【0032】一方、実際の文字のイメージデータは、管理単位メモリ201のデータ部204を副数個使用し格納される。格納されるデータの内容は、最初の2バイトが使用する管理単位メモリ201の個数であり、これに続くデータがイメージデータである。また全てのイメー

ジデータはランレングス圧縮されたデータ列からなる。

【0033】周知の様に、メモリ管理手法としては、可変長のデータ列を、連続したメモリ上に動的に配置する方法が有る。この方法を使用すると、キャッシングへの登録・削除を繰り返した後に、不要となったデータ列による、複数個の、大きさの一定でない空き領域が生じる。このような不要となった断片化したメモリ資源を再配置し、大きな連続した空き領域を再度作り出すためには、メモリのブロック移動を含む実行時間のかかる処理を行う必要がある。これに対し、本実施例の採用した、固定長の管理単位メモリを複数利用する手段は、最終端の管理単位メモリに未使用部分が生じ、メモリ利用率の点で動的メモリ管理に劣るものである。しかし、管理が単純であることから、容易にハードウェア化できる他、ソフトウェアで管理プログラムを実行しても高速処理ができる点で優れている。

【0034】次に、キャッシング割り当て処理207について説明する。

【0035】キャッシングに登録された文字については、

20 そのイメージデータは、キャッシングメモリ205内のどれかの管理単位メモリ201を先頭に複数個の管理単位メモリ201に分割され格納される。このとき先頭となる管理単位メモリ201について、その指標がアドレステーブル208に格納され使用される。本実施例においてはアドレステーブル208は300個の要素からなる配列である。また一つの要素はキーの値209および指標の値210の二つのデータ型からなる。キャッシングメモリ205は約128kバイトの大きさを持ち、先に述べた管理単位メモリ201の1024個の集合である。30 この集合の予め指定された300個は、その指標がアドレス管理テーブル208の各要素に格納されている。ここで、アドレス管理テーブル208に記録される300個の管理単位メモリは、キャッシングメモリ205の先頭から連続に300個指定しても構わない。アドレス管理テーブルの内容は、固定されており、動作中に変更されない。

【0036】本実施例では、300文字までがフォントキャッシングに登録可能であり、この場合、1文字あたりのイメージデータは平均430バイトである。

40 【0037】キャッシング割り当て処理207は、本実施例において処理プログラムとして実現される。図7は、キャッシング割り当て処理207について、その構成を示したものである。キャッシング割り当て処理207は、次の3つの動作をする。

【0038】(1) 外部よりキー212の値を受け取り、イメージデータ206を返す動作（読み出し処理701）。

【0039】(2) 外部よりキー212の値及びイメージデータ206を受け取り、フォントキャッシングへの登録を行なう動作（書き込み処理702）。

【0040】(3) 外部よりキー212の値を受け取り、フォントキャッシュから登録を削除する動作（連結解消処理703）。

【0041】いずれの場合も、キャッシュ割り当て処理207は、キー212の値からアドレステーブル208を参照し、テーブル内のキー値209に対応する目的の指標の値210を取り出し、実アドレス演算部704にて実アドレスに変換し、メモリを指し示すポインタ211を発生する。このポインタ211を使用し、後述する方法で順次キャッシュメモリ205のアドレスバスの内容を指定し、データバス706からイメージデータ206の取り出し、あるいは書き込みが行なわれる。各処理の動作を、以下に説明する。

【0042】(d1-1) データの読み出し
データ読み出しに際し、キャッシュ割り当て処理207は、受け取ったキーを引数として読み出し処理701を呼び出す。読み出し処理701の流れ図を図10に示す。読み出し処理701はキーを取り出し(S101)、アドレス演算部704の発生(S102)したポインタ211に従い(S103)、キャッシュメモリ205内の、文字のイメージデータ格納の先頭を見つけ(S104)、以下連結した管理単位メモリ201への指標に従い文字のイメージデータ206を次々に読み出す(S105)。既に述べた様に、格納されるデータの内容は、最初の2バイトが使用する管理単位メモリ201の個数であるため、必要な個数の管理単位メモリ201を連結することができる。また、一連の連結の最終に位置する管理単位メモリ201の指標部203の内容は-1(16進数でFFFF)である。

【0043】(d1-2) フォントキャッシュからの登録の削除

データの削除された管理単位メモリ201、あるいはデータの空白な管理単位メモリ201は、指標部203の値が0000であることによって他の管理単位メモリ201から区別される。イメージデータの削除に際し、フォント管理機構16は削除すべき文字データに対応するキー212をフォントキャッシュシステム13に渡す。フォントキャッシュシステム13側処理では、このキー212がキャッシュ割り当て処理207に伝達される。キャッシュ割り当て処理207は、連結解消処理703を呼び出す。連結解消処理703の流れ図を図11に示す。連結解消処理703は、さらにアドレス演算部704の呼び出しを行なう(S201)。アドレス演算部704は、キー値209からアドレステーブル208を検索し、削除対象となる文字データの記録されている最初の管理単位メモリ201の指標の値210を取り出す。次に、この指標の値を、演算により実アドレスに変換し、メモリを指し示すポインタ211を発生する(S202)。以下、連結解消処理703は、このポインタが示す管理単位メモリ201の指標部203から次の管理單

位メモリ201の先頭番地の実アドレスを計算し、後に続く指標部203の内容を次々に0000で置き換える(S204)。この操作を管理単位メモリ201の連結が続く限り、指標部203の内容が(16進数で)FFFFである管理単位メモリ201に至るまで繰り返す。このあと、連結の最終の管理単位メモリ201の指標部203の内容を、(16進数で)FFFFから0000に書き換える(S205)。

【0044】(d1-3) フォントキャッシュへのデータの登録

フォントキャッシュへのデータの登録を行なう場合は、キー212と、イメージデータ206が渡され、キャッシュ割り当て処理207によって呼び出された書き込み処理702が実行される。書き込み処理702の流れ図を図12に示す。書き込み処理702は、キー212からキーの値を取り出し(S301)、アドレステーブル208のキーの値209を指定することで、指標の値210を読み取る。この指標の値210をアドレス演算部704で(S302)、イメージデータ書き込みに使用可能な最初の管理単位メモリ201の先頭アドレスへのポインタに変換する(S303)。次に、書き込み処理702は、この管理単位メモリ201にイメージデータの全データサイズを”管理単位メモリのサイズ-2”で割った整数値(2バイト)の内容を書き込む(S304)。これはデータ格納に必要な管理単位メモリ201の総数に一致する。次に、この2バイトのデータに連続して、データ部204の大きさに至るまでのイメージデータを書き込む(S305)。もし更に、イメージデータの残りバイト数があれば、書き込み処理はこのデータをまずあふれ領域213に書き込み(S306)、キャッシュメモリ205内部で、未使用の管理単位メモリ201を捜す(S307)。もし未使用的管理単位メモリ201が発見できれば、あふれ領域213から、126バイト(これは、管理単位メモリ201のデータ部204のサイズである)のデータが取り出され書き込まれる(S305)。さらに、先にアクセスした管理単位メモリ201の指標部203に、現在の管理単位メモリ201への指標を書き込み、新たな結合を生成する(S308)。これを、あふれ領域213の内容が空になるか、未使用的管理単位メモリ201が無くなるまで繰り返す。あふれ領域213の残りデータが無くなった場合、書き込み処理は正常終了する。この場合、最後に使用した管理単位メモリ201の指標部203の内容を、結合終了を示す値FFFFに書き換え、結合の終端を生成する(S309)。

【0045】一方、未使用的管理単位メモリ201が必要な数だけ検出できず、あふれ領域213にイメージデータが残っている場合は、書き込み処理702は、呼び出し側であるキャッシュ割り当て処理207に対し、書き込みエラーを返し(S310)処理を中止する。この

場合、不要な文字のデータをフォントキャッシュメモリ 205 から取り除き、空き領域を広げ、再度書き込み処理 702 を呼び出す必要があるが、この処理は、フォント管理機構 12において判断、実行される。処理の従属関係は、フォント管理機構 12 の登録処理 602 が、キャッシュ割り当て処理 207 を呼び出し、更にキャッシュ割り当て処理 207 が、書き込み処理 702 を呼び出す関係である。

【0046】あふれ領域 213 は、1 文字のデータには充分大きな容量を持つ連続したアドレスのメモリ領域であるが、もしあふれ領域 213 に入りきらない大きさの文字のイメージデータが渡された場合は、書き込み処理 702 はあふれ領域エラーを返し (S311) 処理を中止する。この場合、この文字のイメージデータはフォントキャッシュに登録されない。なぜなら、特に大きなイメージデータを持つ文字は、発生頻度が極めて低いため、キャッシュに登録しても、システムの処理速度改善に寄与しないためである。

【0047】(d2) フォントサーバ側のフォント管理機構

図8はフォントサーバ側のフォント管理機構 16 の構成図である。

【0048】フォント管理機構 16 は、要求側コンピュータ 1 から、ネットワークドライバ 14 を介し、文字取り出し要求 22、文字登録要求 21 のどちらかを受け取る。既に述べた様に、ネットワークでのデータの送信、受信はネットワークの仕様に従ったパケットで行なわれる。フォントサーバ 2 側は、受け取ったパケットのデータ列から、送信側コンピュータのネットワーク・アドレスを取り出し、要求を発生したコンピュータ 1 を特定することができる。

【0049】フォントサーバ 2 側は、文字取り出し要求 22 と文字指定レコード 20 を受け取った場合、文字指定レコード 20 の内容に従い、文字のイメージデータ 206 を発生し、要求を発生したコンピュータ 1 に渡す。

【0050】文字登録要求 21、文字指定レコード 20、文字のイメージデータ 206 を受け取った場合は、文字指定レコード 20 の内容をフォント管理テーブル 805 に書き込み、文字のイメージデータ 206 をフォントキャッシュシステム 15 に登録する。

【0051】以下、それぞれの動作を説明する。また、フォント管理テーブル 805 の一つの要素の構成を図9に示す。フォント管理テーブルのキー 808 の内容は、初期状態において、フォントキャッシュシステム 15 のアドレステーブル 208 のキーの内容 209 と一致する様に設定され、その後変更されない。

【0052】(d2-1) 文字取り出し要求 22 を受信した場合の動作

文字取り出し要求 22 を受け取ったフォント管理機構 16 は、読み出し処理 801 を起動する。読み出し処理 8

01 の流れ図を図13に示す。読み出し処理 801 は、テーブル検索処理 804 を呼び出し (S401)、文字指定レコード 20 の内容と同一の要素が、フォント管理テーブル 805 に登録されているか検索する。

【0053】まず同一の要素が検出された場合について書く。

【0054】この場合は、アクセスカウンタ 809 を +1 する (S402)。アクセスカウンタ 809 は 48 ビット長のカウンタで、システム立ち上げ時にクリアされ、以降クリアされず、フォント管理テーブル 805 で、検索対象が検出された場合と、フォント管理テーブル 805 に新たに登録された要素が有った場合、+1 される。本実施例の使用状態では、オーバーフローは発生しない。アクセスカウンタ 809 の値は、検出された要素に対応した登録フラグ 806 に記録される (S403)。アクセスカウンタ 809 の値は、登録、読み出しが行なわれた回数をカウントするため、この数が大きい程、最近アクセスされた要素であると判断できる。続いて、検出された要素に対応したキー 808 が、読み出される。読み出し処理 801 は、このキー 212 をフォントキャッシュシステム 15 に渡し、読み出し処理を指示し (S404)、処理結果として文字のイメージデータ 206 を得る。フォント管理機構 16 は、このイメージデータ 206 を、ネットワークドライバ 14 を介して、要求側コンピュータ 1 に転送し (S405)、処理を終了する。

【0055】一方、文字指定レコード 20 で指定された文字が、フォント管理テーブル 805 に登録されていなかった場合の動作について書く。

【0056】この場合、読み出し処理 801 は、文字指定レコード 20 を、フォント発生機構 17 に渡し (S406)、処理結果としてイメージデータ 206 を取得する。以下の処理は、前述同様であり、フォント管理機構 16 は、このイメージデータ 206 を、ネットワークドライバ 14 を介して、要求側コンピュータ 1 に転送し (S405)、処理を終了する。

【0057】(d2-2) 文字登録要求 21 を受信した場合の動作

文字登録要求 21 を受信した場合、フォント管理機構 16 は、登録処理 802 を起動する。登録処理 802 の流れ図を図14に示す。登録処理 802 は、テーブル検索処理 804 を呼び出す (S501)。文字指定レコード 20 に関し検索し、既に同じデータが登録されていれば、処理はただちに終了する。次に、フォント管理テーブル 805 の登録フラグ 806 を検索し (S502)、値の最も小さい要素を取り出す。登録フラグには、登録時あるいは読み出し時のアクセスカウンタ 809 が書き込まれているため、この値が小さいとしたら、その登録フラグを持つ文字データは、最近使用されていないものと判断できる。本実施例は、削除する文字の決定に際し、この

登録フラグ806の値の最も小さいものを選ぶ。この判断方法は前述のLeast Recently Used(LRU)の方法である。読み出した登録フラグ806の値が0であれば、その要素は、今までフォント登録の対象とならなかつたか、あるいは削除されたばかりの要素であると判断する。そこで、登録フラグ806の内容を16進数のFFFFFFFFFFに書き換える。これは現在使用中であることを示す一時的フラグである。続いて、この要素に対応するキー808の内容を取り出し、キー212とイメージデータ206をフォントキャッシュシステム15に転送し、書き込み処理を指示する(S503)。処理結果の戻り値として、フォントキャッシュシステム15から、あふれ領域エラーが戻された場合は、何も行なわず、登録処理を終了する。なぜなら(d1-3)で述べた様に、登録に値しないデータであると判断されるからである。またこの場合、登録処理終了時には、登録フラグ806の値を、処理前の値に復元する。

【0058】 処理結果の戻り値として、フォントキャッシュシステム15から、書き込みエラーが戻された場合は、登録処理802は削除処理803を呼び出す(S504)。削除処理803は、テーブル検索処理804を使用し、フォント管理テーブル805の登録フラグ806を検索し、値が0以外で最も小さい要素を取り出す。次にこの要素に対応したキー808の内容を取り出し、キー212に代入しフォントキャッシュシステム15にキャッシュからの削除を指示する。削除処理803はこの操作で処理を終了し、制御を登録処理802に戻す。この処理の結果、フォントキャッシュシステム15には空き領域が生じるため、登録処理802は再び、キー212とイメージデータ206をフォントキャッシュシステム15に転送し、書き込み処理を指示する。更に、処理結果の戻り値として、フォントキャッシュシステム15から、書き込みエラーが戻された場合は、同様の処理を繰り返す。あふれ領域エラーが発生しない程度の大きさのイメージデータであれば、少なくとも何回かの処理の繰り返しによって、登録が完了する。書き込みが、エラー無く終了した場合は、登録処理終了時にアクセスカウンタ809の内容に+1を行ない(S505)、この値を登録フラグ806に書き込む(S506)。

【0059】 フォント管理テーブル805の登録フラグ806を検索し(S502)、フラグの値の最も小さい要素を取り出す処理で、フラグの値が0以外であった場合は、連結解消処理703を呼び出し(S507)、フォントキャッシュシステム15から、文字データを削除し、書き込み処理をS502から継続する。登録が完了した場合は、フォント管理機構16は正常終了の処理結果を戻し、フォントサーバ2は、ネットワークドライバ14を介し、要求側コンピュータに処理終了のメッセージを返送する。

【0060】

(d3) 文字要求側コンピュータのフォント管理機構
図6は文字要求側コンピュータ(クライアント)のフォント管理機構12の構成図である。

【0061】 アプリケーション11は、文字指定レコード20をフォント管理機構12に渡し、処理結果として文字のイメージデータ206を受け取る。一方、フォント管理機構12は、フォントキャッシュシステム13に登録された文字であれば、イメージデータ206をフォントキャッシュシステム13から取り出し、それ以外10は、ネットワークドライバ14を使用して、ネットワーク上のフォントサーバからイメージデータ206を取得する。以下、動作の詳細を説明する。

【0062】 フォント管理機構12の主たる構成要素は、読み出し処理601、登録処理602、削除処理603、フォント管理テーブル605である。各処理の動作は次の通りである。また、フォント管理テーブル605の一つの要素の構成を図9に示す。

【0063】 (d3-1) 削除処理

削除処理603は、登録処理602により呼び出され使用される。この処理の流れ図を図5に示す。文字出力要求側のコンピュータ1(すなわちクライアント)のフォント管理機構12において、削除処理603は、フォントサーバ2側のフォント管理機構16の削除処理803と大きく異なる。削除処理603は、テーブル検索処理604を使用し、フォント管理テーブル605の登録フラグ606を検索し(S601)、値が0以外で最も小さい要素を取り出す(S602)。次にこの要素に対応したキー608の内容と、文字指定レコード607の内容を読み出す(S603)。続いて、まずキー608の内容をキー212に代入しフォントキャッシュシステム13にキャッシュからの読み出しを指示し(S604)、処理結果としてイメージデータ206を取得する(S605)。さらに、削除処理603は、文字指定レコード607から読み出した内容を、文字指定レコード20に代入し、フォントキャッシュシステム13から読み出されたイメージデータ206と合わせ、文字登録要求21を生成し、ネットワークドライバ14を介し、フォントサーバ2へ送信する(S606)。この後、キー212を指定し、フォントキャッシュシステム13に削除を指示する(S607)。

【0064】 (d3-2) 登録処理

登録処理602は、読み出し処理601からイメージデータ206と、文字指定レコード20を引数として呼び出され使用される。この処理の流れ図を図4に示す。登録処理602は、テーブル検索処理604を呼び出し(S701)、フォント管理テーブル605の登録フラグ606を検索し、値の最も小さい要素を取り出す(S702)。このとき、登録フラグ606の値が0であれば、その要素は、今までフォント登録の対象とならなかつたか、あるいは削除されたばかりの要素であると判断50

する。そこで、登録フラグ606の内容を16進数のFFFFFFFFに書き換える。これは現在使用中であることを示す一時的フラグである。続いて、この要素に対応するキー608の内容を取り出し、キー212とイメージデータ206をフォントキャッシュシステム13に転送し、書き込み処理を指示する(S703)。処理結果の戻り値として、フォントキャッシュシステム13から、あふれ領域エラーが戻された場合は、何も行なわず、登録処理を終了する。この場合、登録処理終了時には、登録フラグ606の値を、処理前の値に復元する。

【0065】処理結果の戻り値として、フォントキャッシュシステム13から、書き込みエラーが戻された場合は、登録処理602は削除処理603を呼び出す(S704)。この処理の結果、フォントキャッシュシステム13には空き領域が生じるため、登録処理602は再び、キー212とイメージデータ206をフォントキャッシュシステム15に転送し、書き込み処理を指示する。更に、処理結果の戻り値として、フォントキャッシュシステム13から、書き込みエラーが戻された場合は、同様の処理を繰り返し、登録が完了する。この場合は、登録処理終了時にアクセスカウンタ609の内容に+1を行ない(S705)、この値を登録フラグ606に書き込む(S706)。

【0066】フォント管理テーブル605の登録フラグ606を検索し、値の最も小さい要素を取り出し(S702)たとき、この値が0でなければ、削除処理603を呼び出し(S707)、開き領域を確保し、S702以降の処理を行なう。

【0067】(d3-3) 読み出し処理

読み出し処理601は、フォント管理機構12の主たる処理である。流れ図を図3に示し、説明する。フォント管理機構12は、文字指定レコード20の内容を引数に読み出し処理601を実行する。読み出し処理601は、テーブル検索処理604を呼び出し(S801)、文字指定レコード20の内容と同一の要素が、フォント管理テーブル605に登録されているか検索する。

【0068】まず同一の要素が検出された場合について書く。

【0069】この場合はアクセスカウンタ609の内容を+1する(S802)。アクセスカウンタ609は、809と同様の構成である。アクセスカウンタ609の値は、検出された要素に対応した登録フラグ606に記録される(S803)。続いて検出された要素に対応したキー608が読み出される(S804)。読み出し処理601は、このキー608の値をキー212に代入し、フォントキャッシュシステム13に渡し、読み出し処理を指示し(S805)、処理結果として文字のイメージデータ206を得る。フォント管理機構12は、このイメージデータ206をアプリケーション11に渡し処理を終了する。

【0070】一方、文字指定レコード20で指定された文字が、フォント管理テーブル605に登録されていなかった場合の動作について書く。この場合、読み出し処理601は、文字取り出し要求22のパケットをネットワーク上のフォントサーバ2へ送信する(S806)。応答パケットとして、文字のイメージデータ206を含むパケットが受信されたら(S807)、読み出し処理601はこのイメージデータ206をアプリケーション11に渡す。次に、文字指定レコード20、およびこのイメージデータ206を引数に登録処理602を呼び出す(S808)。登録処理602は、すでに述べた処理を実施し、この結果、フォントキャッシュシステム13の更新が行なわれ、読み出し処理601は終了する。

【0071】

【発明の効果】本発明によれば、次の様な効果を得ることができる。まず、ネットワーク上のフォントサーバを使用することから生じる効果として次の点が指摘できる。

【0072】(1) 要求側装置に、フォント元データを持つ必要が無いため、補助記憶装置などの資源の有効利用が図れる。

【0073】(2) 供給装置側で、フォント元データを一元管理することにより、新規の文字種の追加などの処理を容易に行なうことができる。

【0074】(3) 供給側装置によって、各文字のイメージデータ生成を行なうため、供給側装置を専用機とすることで、処理上特殊な操作を行なう必要のある文字データ等への対応が容易である。

【0075】さらに、フォントサーバ(供給装置)側にフォントキャッシュ機構を持つことによって、次の効果が指摘できる。

【0076】(4) 要求側装置で、文字種の変更などによって、ローカルなフォントキャッシュのヒット率が悪化した場合でも、他の装置の稼働により、フォントサーバ上のフォントキャッシュには、種々の文字が登録されており、グローバルなキャッシュから要求文字を引き当てることができる確率が高い。

【0077】(5) フォントサーバの処理が、フォントサーバ側にフォントキャッシュを実装したことで向上し、結果的にシステム全体の処理速度の向上に結びつく。

【0078】これに加えて、本発明が要求装置上のローカルなフォントキャッシュの内容と、フォントサーバ上のグローバルなフォントキャッシュの内容を不一致に保つ操作を行なうため、次の様な効果が得られる。

【0079】(6) 要求側装置のフォントキャッシュと、フォントサーバ上のフォントキャッシュにおいて、同一文字が重複し登録される率が低く、有限なキャッシュメモリに、多くの文字を効率良く登録できる。

【図面の簡単な説明】

【図1】一つの実施例の概略構成図。

【図2】フォントキャッシュシステムの構成図。

【図3】要求側装置での、読み出し処理の流れ図。

【図4】要求側装置での、文字データ登録処理の流れ図。

【図5】要求側装置での、文字データ削除処理の流れ図。

【図6】要求側装置のフォント管理機構の説明図。

【図7】キャッシング割り当て処理の説明図。

【図8】フォントサーバ側のフォント管理機構の説明図。

【図9】フォント管理テーブルの要素の説明図。

【図10】キャッシング管理処理での、データ読み出し処理の流れ図。

【図11】キャッシング管理処理での、連結解消処理の流れ図。

【図12】キャッシング管理処理での、データ書き込み処理の流れ図。

【図13】フォントサーバでの文字データ読み出し処理の流れ図。

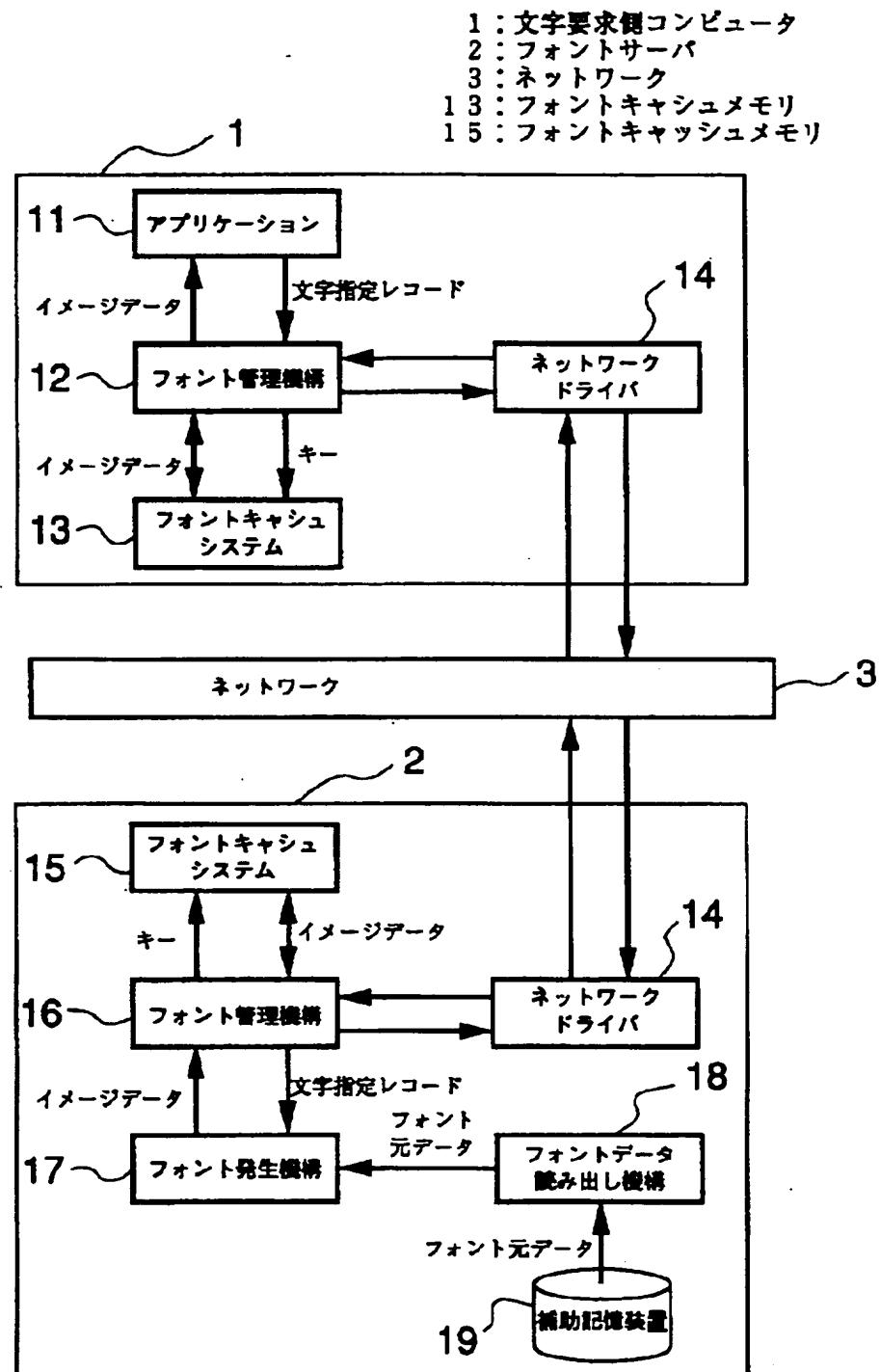
【図14】フォントサーバでの文字データ登録処理の流れ図。

【符号の説明】

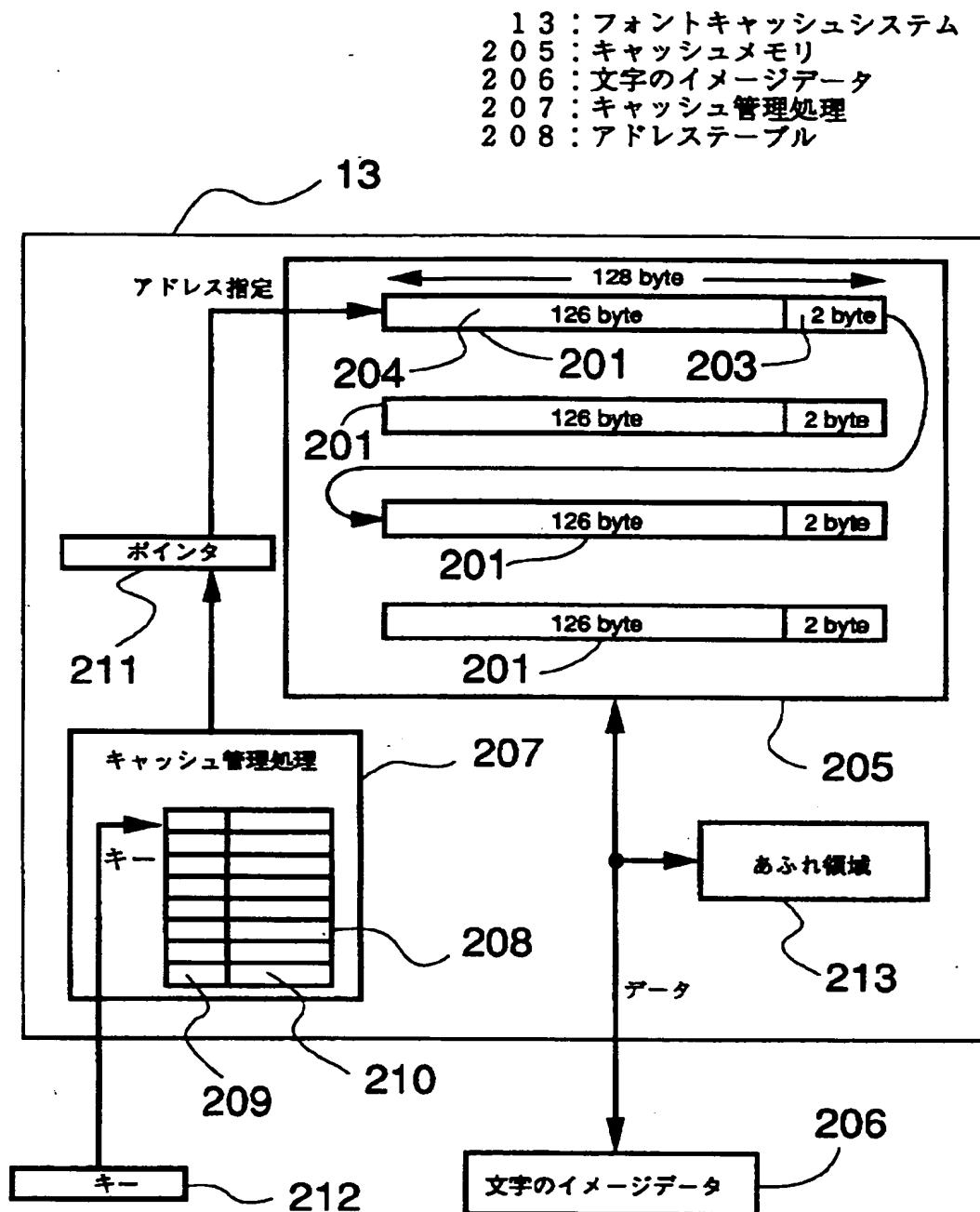
1…文字要求側コンピュータ
2…フォントサーバ
3…ネットワーク
1 1…アプリケーション
1 2…フォント管理機構
1 3…フォントキャッシュシステム
1 4…ネットワークドライバ
1 6…フォント管理機構（フォントサーバ側）

1 7…フォント発生機構
1 8…フォントデータ読み出し機構
1 9…補助記憶装置
2 0…文字指定レコード（変数）
2 0 1…管理単位メモリ
2 0 3…指標部
2 0 4…データ部
2 0 5…キャッシングメモリ
2 0 6…文字のイメージデータ
10 2 0 7…キャッシング管理機構
2 0 8…アドレステーブル
2 0 9…キーの値
2 1 0…指標の値
2 1 1…ポインタ
2 1 2…キー（キーの値を格納する変数）
6 0 1、8 0 1…読み出し処理
6 0 2、8 0 2…登録処理
6 0 3、8 0 3…削除処理
6 0 4、8 0 4…テーブル検索処理
20 6 0 5、8 0 5…フォント管理テーブル
6 0 6、8 0 6…登録フラグ
6 0 7、8 0 7…文字指定レコード
6 0 8、8 0 8…キー
6 0 9、8 0 9…アクセスカウンタ
7 0 1…読み出し処理
7 0 2…書き込み処理
7 0 3…連結解消処理
7 0 4…実アドレス演算部
7 0 5…アドレスバス
30 7 0 6…データバス

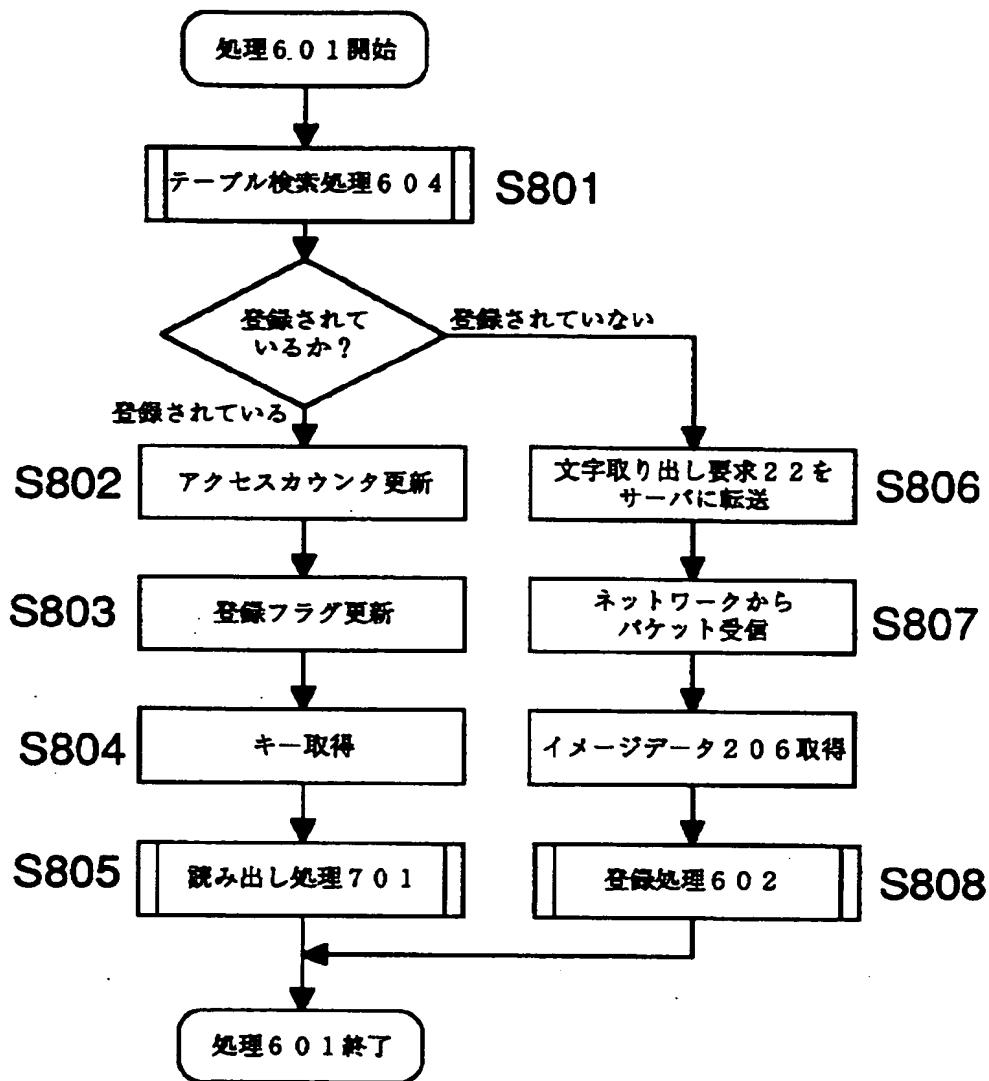
【図1】



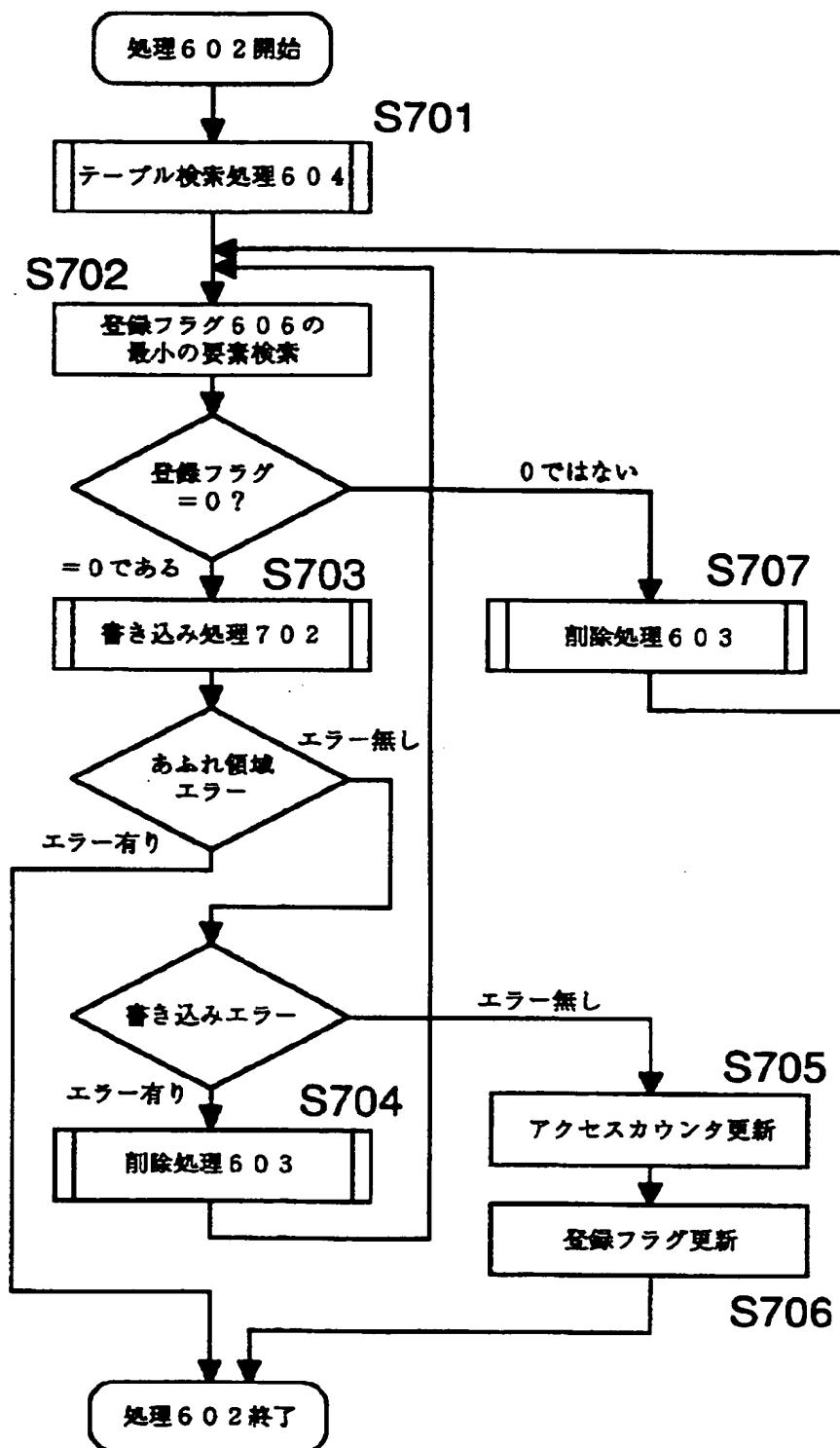
【図2】



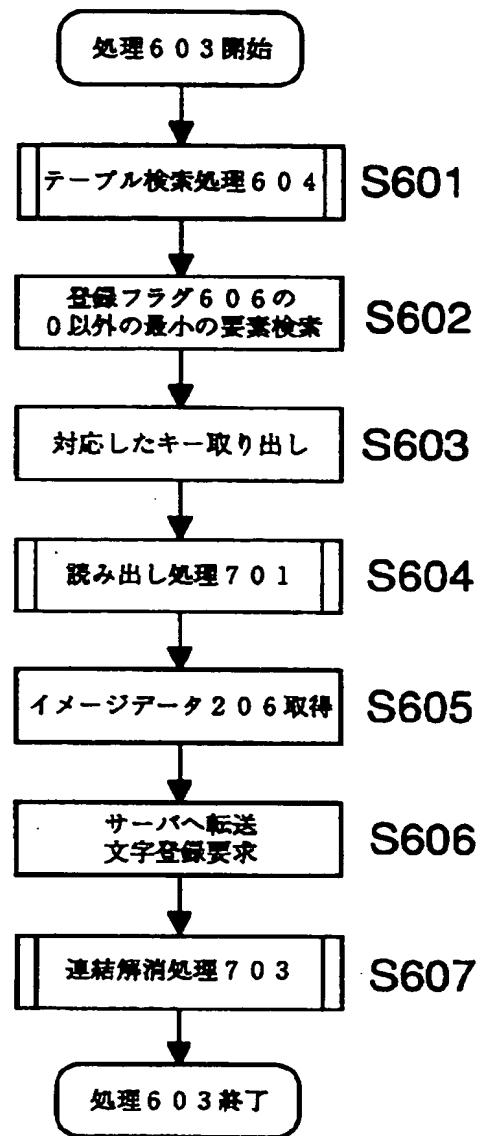
【図3】



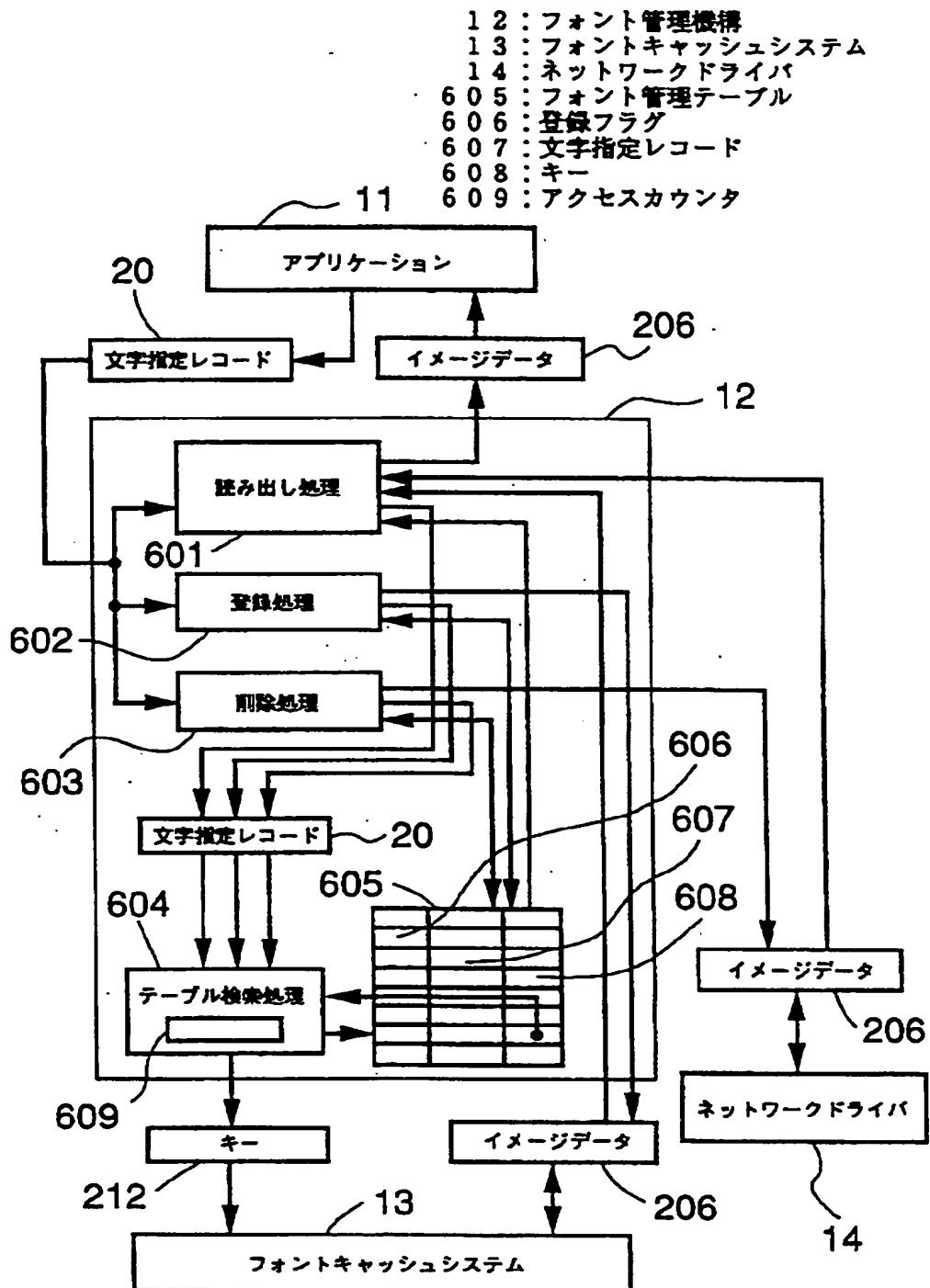
【図4】



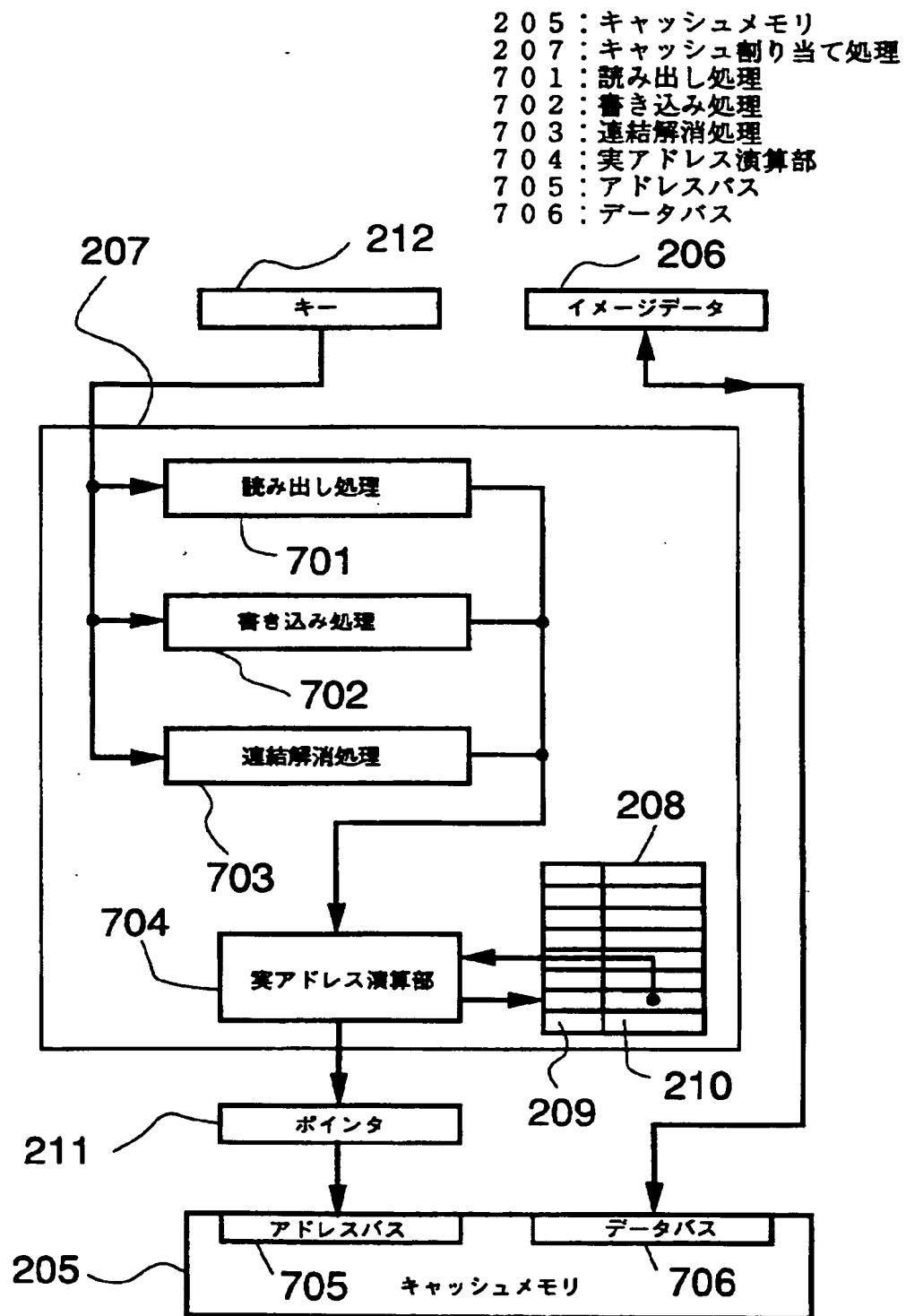
【図5】



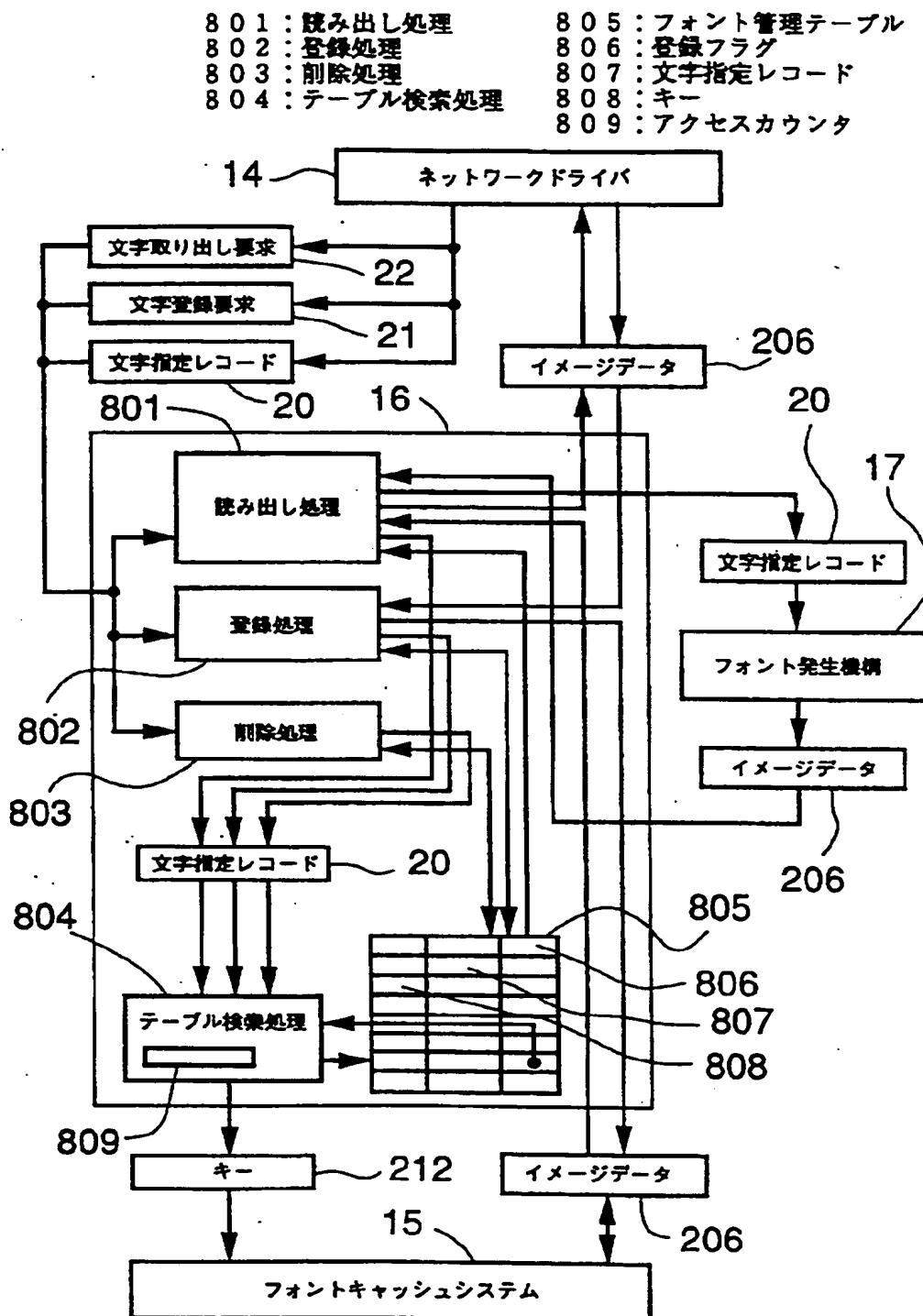
【図6】



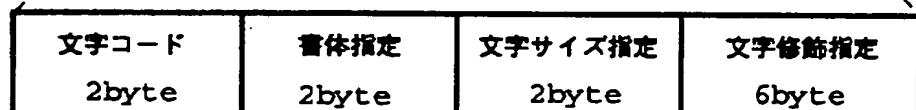
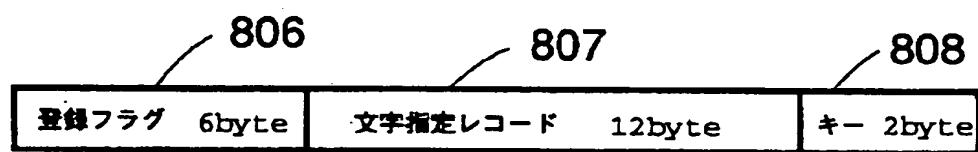
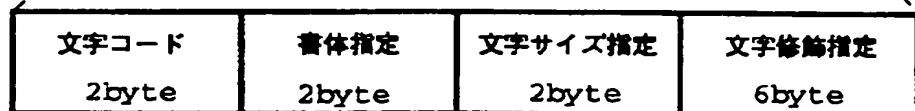
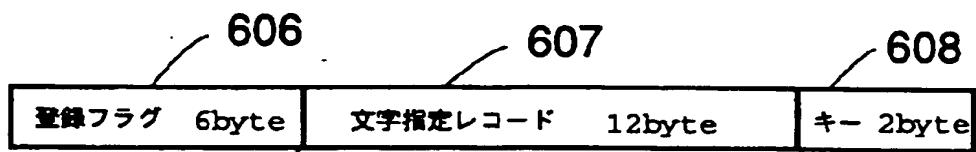
【図7】



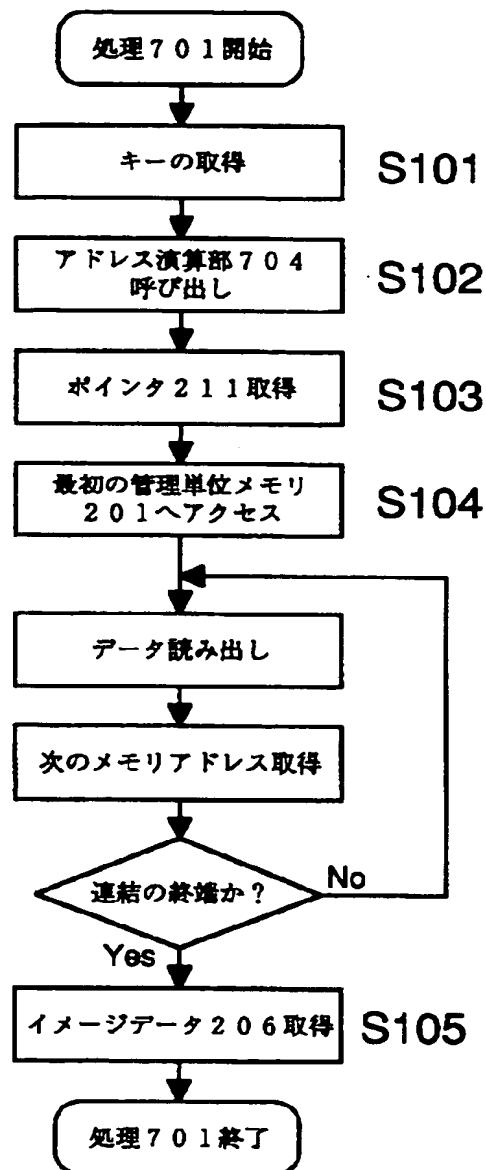
[図 8]



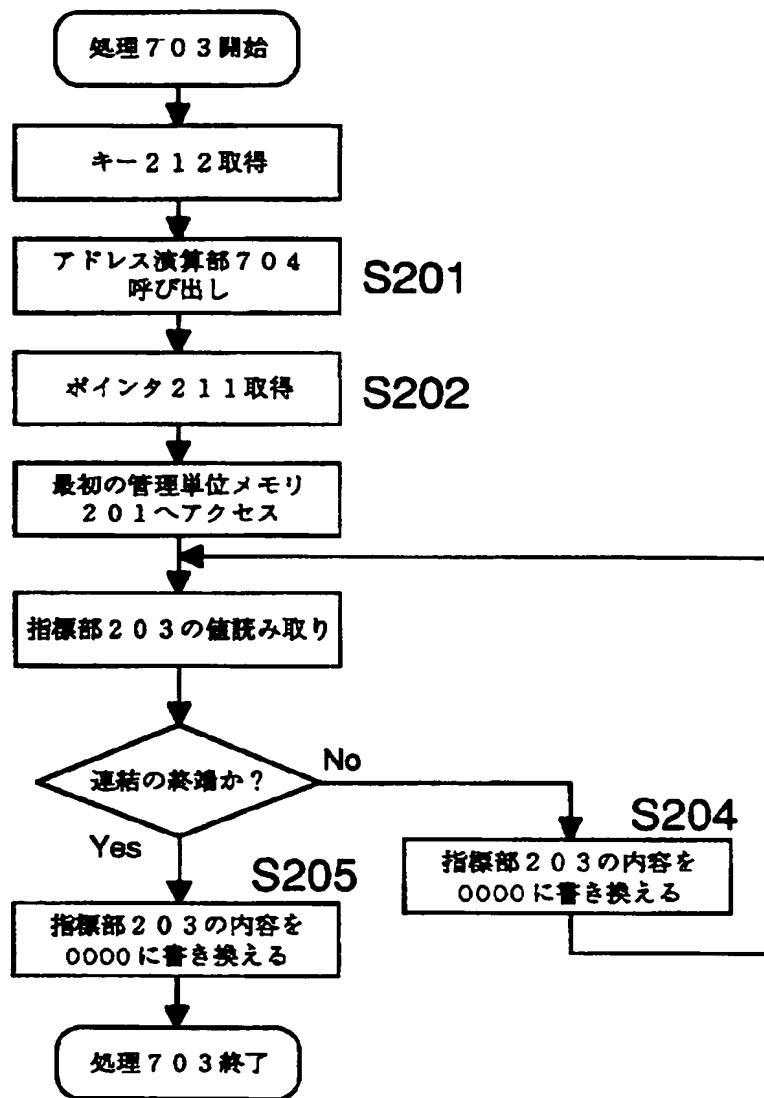
【図9】



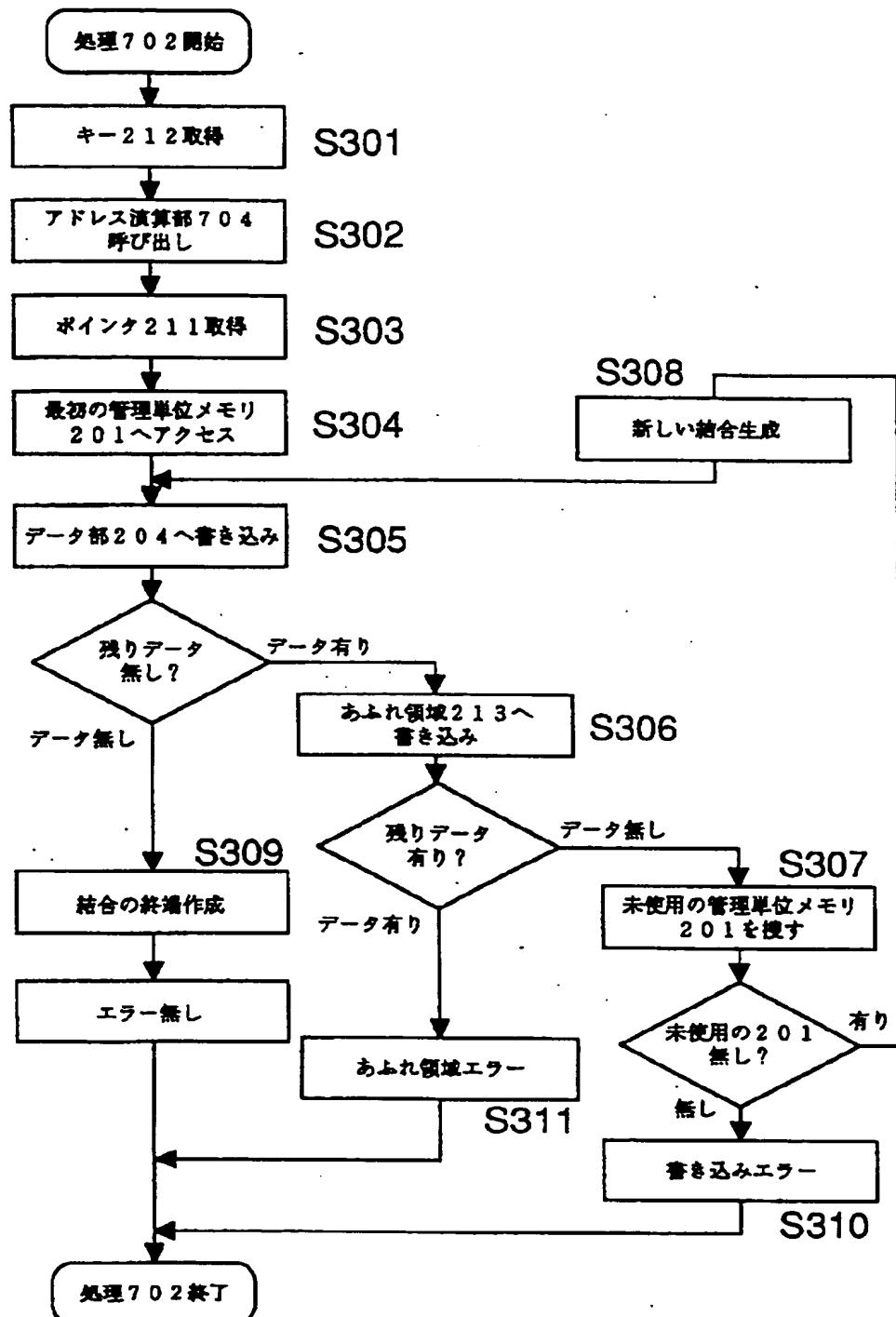
【図10】



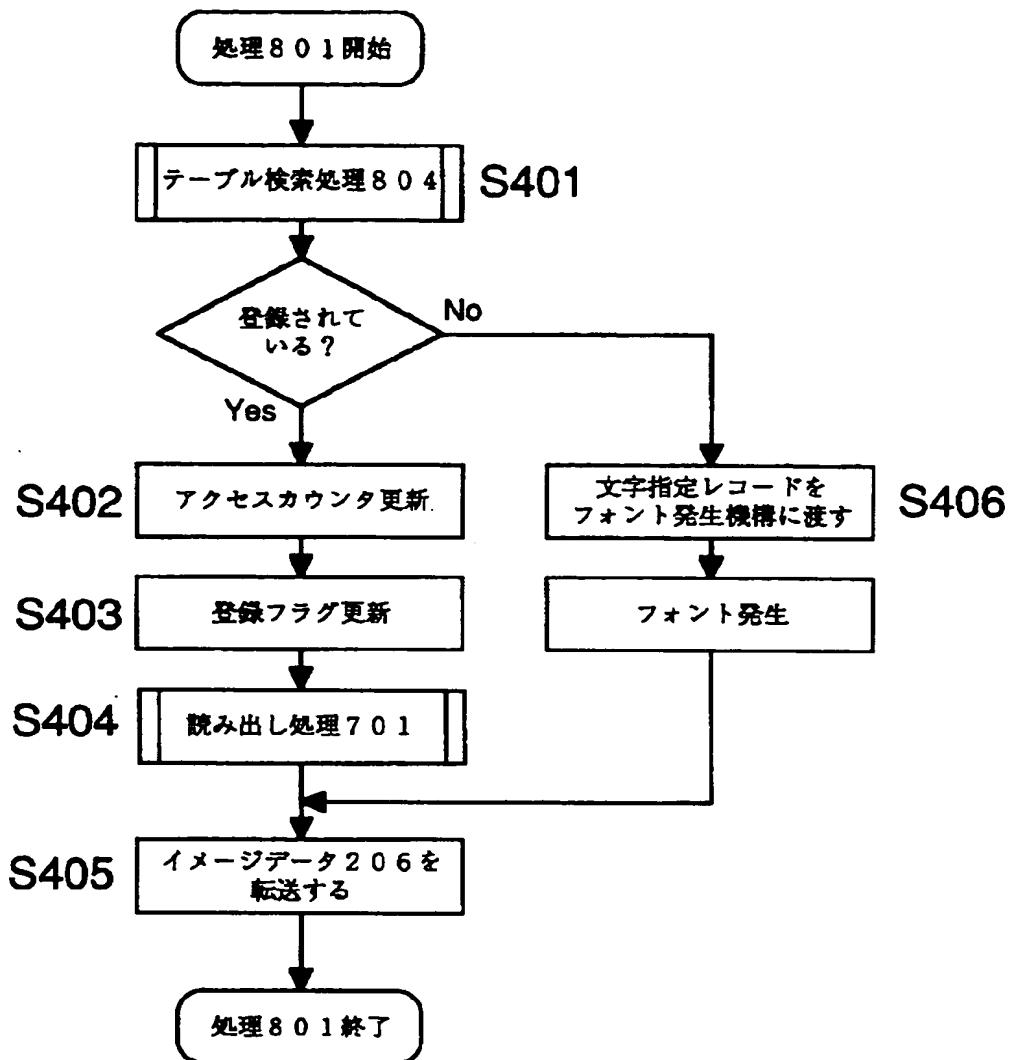
【図11】



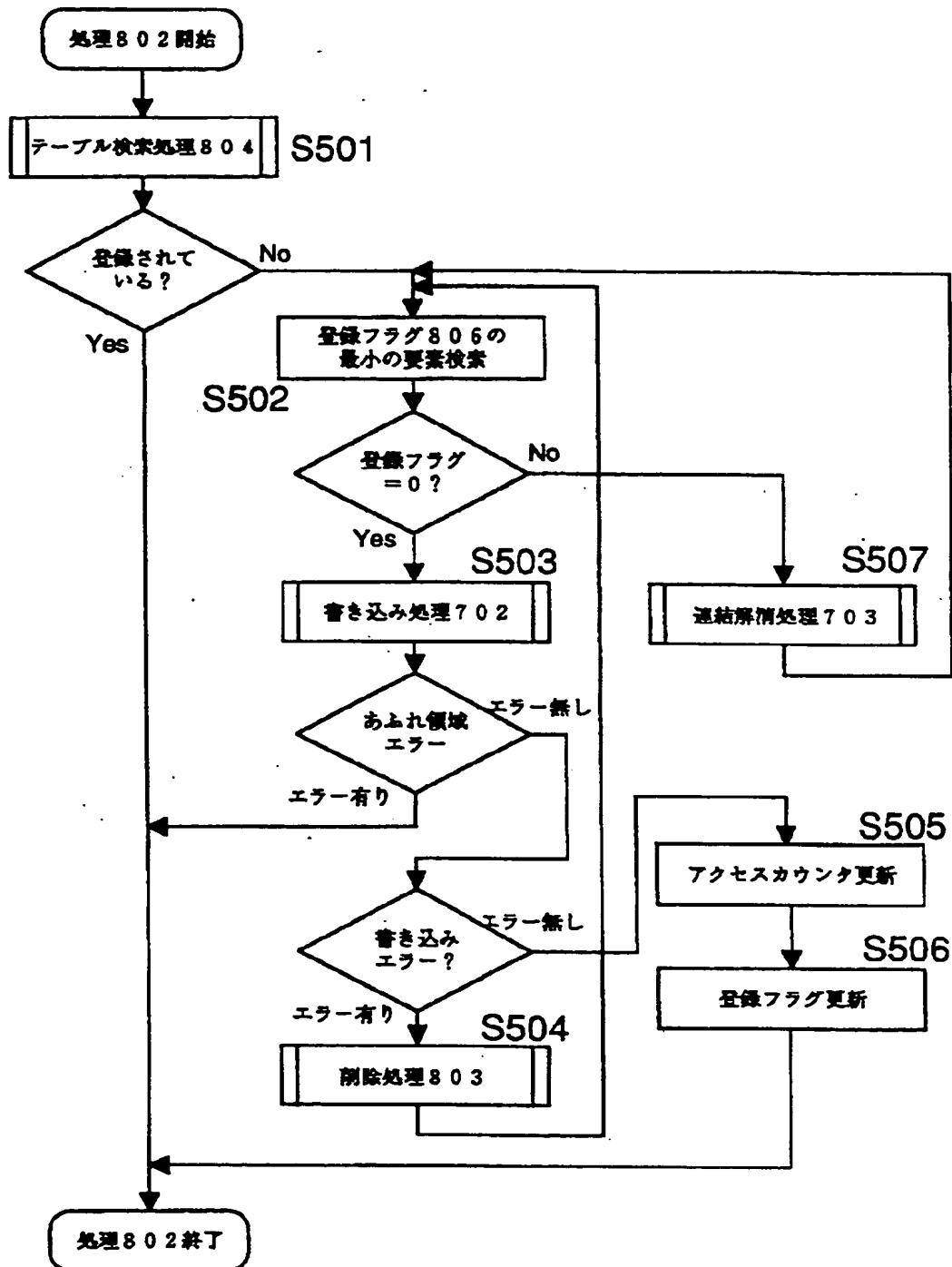
【図12】



【図13】



[図 14]



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.